UCRL-TR-233689

HPC Colony: Linux at Large Node Counts

T. Jones, A. Tauferner, T. Inglett, A. Sidelnik

August 14, 2007

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

HPC Colony: Linux at Large Node Counts
Report from Experiments Conducted on Sixth BGW Day

Terry Jones, Lawrence Livermore National Laboratory
Andrew Tauferner, IBM Rochester
Todd Inglett, IBM Rochester
Albert Sidelnik, IBM Rochester

August 10, 2007

1. Introduction

As part of the HPC-Colony project sponsored by the Department of Energy's FastOS program (http://www.hpc-colony.org), we are conducting research on OS issues on extremely large systems. [Chakravorty06] In particular, we are actively investigating scaling issues associated with system software including certain coordination aspects of our smart runtime system and operating system, operating system jitter, and administrative activities. [Jones03a, Jones03b] During the recent BGW-Day, we had a chance to test the scalability of some of our approaches in those areas. We were able to start our tests on configurations with a few thousand BG/L processors, and increase the test configuration to the full machine size, so that the scaling behavior of our techniques can be assessed. This document describes the tests we conducted, the results we obtained, and the conclusions that we could draw from such experiments.

The remainder of this report is organized as follows. In Section 2 we briefly describe our motivation. Section 3 describes the experiments that we conducted and provides the scalability results on large BGW configurations. Finally, Section 4 contains our conclusions and a summary of our achievements during BGW-Day.

2. Motivation

Our tests were designed to measure the performance of our current Linux kernel. Future tests are planned for collecting data on co-scheduling and lock optimizations. For this round of tests, we conducted an assessment of the feasibility of our schemes on machines with many thousands of processors. We envision a computational environment where full Linux is one of a number of choices for compute node system software, thus realizing a portable environment for large node count supercomputers. [Jones07] In order to realize this vision, the boot times for Linux on extreme scale diskless systems must not be prohibitive.

Furthermore, the noise usually attributed to full Linux must not be prohibitive. Operating system jitter may stem from many sources including timer decrement interrupts, tlb activity, and interference from daemons and threads other than the parallel application running on the system. Colony mitigates the negative impact through a variety of techniques including a large-page tlb design and parallel aware co-scheduling. The tests we performed on July 26th will provide a baseline for strategies we are developing to address excellent portability together with excellent scalability. Our tests collected valuable scaling data for our “Linux on the compute node” approach.

3. Results

3.1 Experiment Description

First, we collected scaling times for each step of booting full Linux at scale and mounting file systems. This was accomplished with multiple runs of various partition sizes, and analyzing the console logs for the various boot steps. For comparison purposes, we also collected data on IBM’s standard lightweight BlueGene/L kernel known as CNK.

Second, we measured the impact and overhead of our big-page Linux through LLNL developed benchmarks centered on all-reduce performance. These comparisons allow us to quantify the performance penalty imposed by the Colony Linux kernel and associated sources of interference (e.g. tlb activity, daemons, and so forth). We have collected a wealth of data and are currently working through the data to understand our timings and measurements. The source of performance penalties is especially important, though not always easy to find. Our real-machine measurements and analysis will help us to further optimize our system software stack.

The measurement tool used to help us quantify scaling performance was Presta Glob version 1.4.3. This test, also referred to as the Global-Op benchmark, times MPI operation loops with and without a simulated compute phase for MPI_Barrier, MPI_Reduce, MPI_Bcast, MPI_Allreduce. Our experiments focused on MPI_Allreduce. Timing results are obtained in each of these tests through the use of the MPI_Wtime function. To address variation in MPI_Wtime resolution, all of the tests, with the exception of globalop, allow the specification of the number of operations to be performed between MPI_Wtime calls as a command-line argument. The number of MPI_Wtime clock ticks per measurement is provided in the output of each test. This mechanism is provided as a direct means of tuning the test behavior to the granularity of the MPI_Wtime function.

3.2 Boot Times

The compute nodes of BlueGene/L machines are diskless. Much care has been given to ensure that boot times are not prohibitive, and boot images are able to utilize a high-speed interconnect between IO nodes and compute nodes. As stated previously, our usage model depends on the ability to select full Linux for portability without having to sacrifice scaling performance. Practical concerns dictate that even small execution jobs (such as those associated with the debugging/trial phase) will not have prohibitive boot times compared to other options.

Our results show that there is fair amount of variation for the CNK lightweight kernel (see Figure 1). Times of both 1 minute 52 seconds and 7 minutes 45 seconds were measured for partitions of size 8256 nodes. The big-pages Linux boot times scaled logarithmically (e.g., appeared linear on a semi-log plot) ranging from around 4 minutes at 1024 nodes to 15 minutes and 44 seconds on 20,480 nodes.

This compares to 2:50 seconds for a typical Linux Opteron disk-full node (whopper.llnl.gov), and 3 minutes and 18 seconds for a typical AIX Power5+ disk-full node (purple011.llnl.gov).

For clusters comprised of diskless compute nodes, the boot time will depend on the fabric and IO server providing the boot image. At LLNL, each 144 nodes of the Peloton systems are separate instances which may be booted totally in parallel without any common hardware or software points. Each set of 144 nodes boots in approximately 11.5 minutes (zeus.llnl.gov).

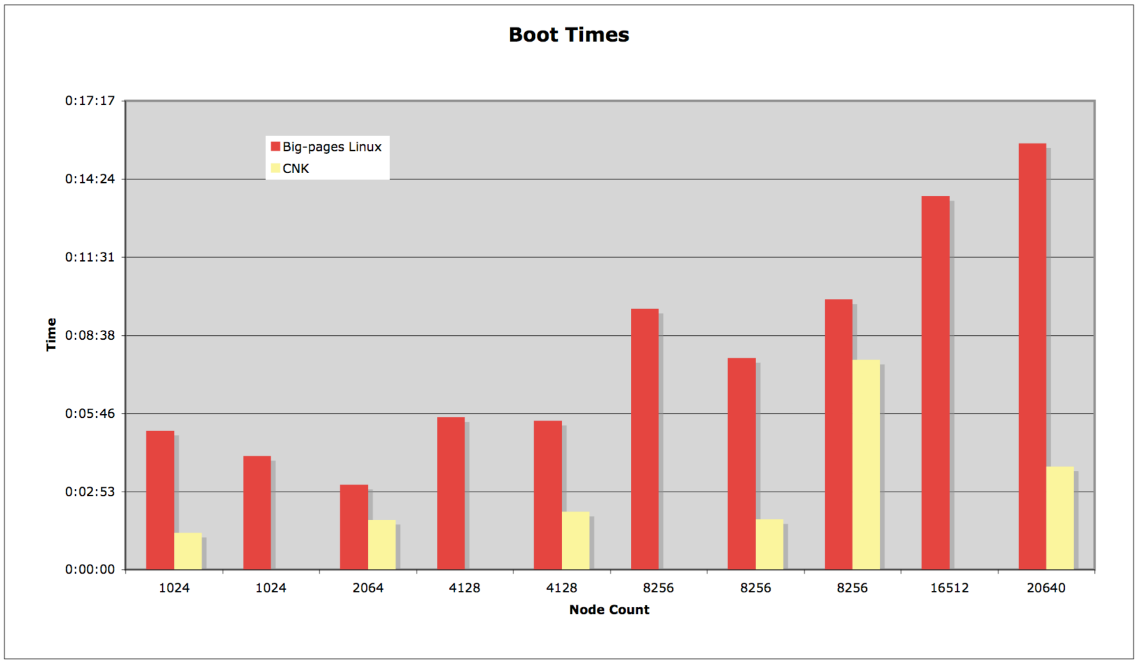

Figure 1: Boot times on a 20-rack BlueGene system

3.3 Allreduce Scaling

Synchronizing collective operations are operations in which a set of processes (frequently every process) participates and no single process can continue until every process has participated. Examples of synchronizing collective operations from the MPI interface are MPI_Barrier, MPI_Allreduce, and MPI_Allgather. These operations pose serious challenges to scalability since a single instance of a laggard process will block progress for every other process. Unfortunately, synchronizing collective operations are required for a large class of parallel algorithms and are quite common. [Gupta91]

A *cascading effect* results when one laggard process impedes the progress of every other process. The cascading effect has significant operating system implications and it proves especially detrimental in an HPC context: while operating systems may be considered very efficient in a serial context, even minimal system and/or daemon activity proves disastrous due to the cascading effect in the large processor count parallel environment common in HPC centers.

Our results for Big-pages Linux show very good allreduce scaling for today's standards (e.g. compared to the ASC Purple machine), but substantially below the CNK kernel (see Figure 2).

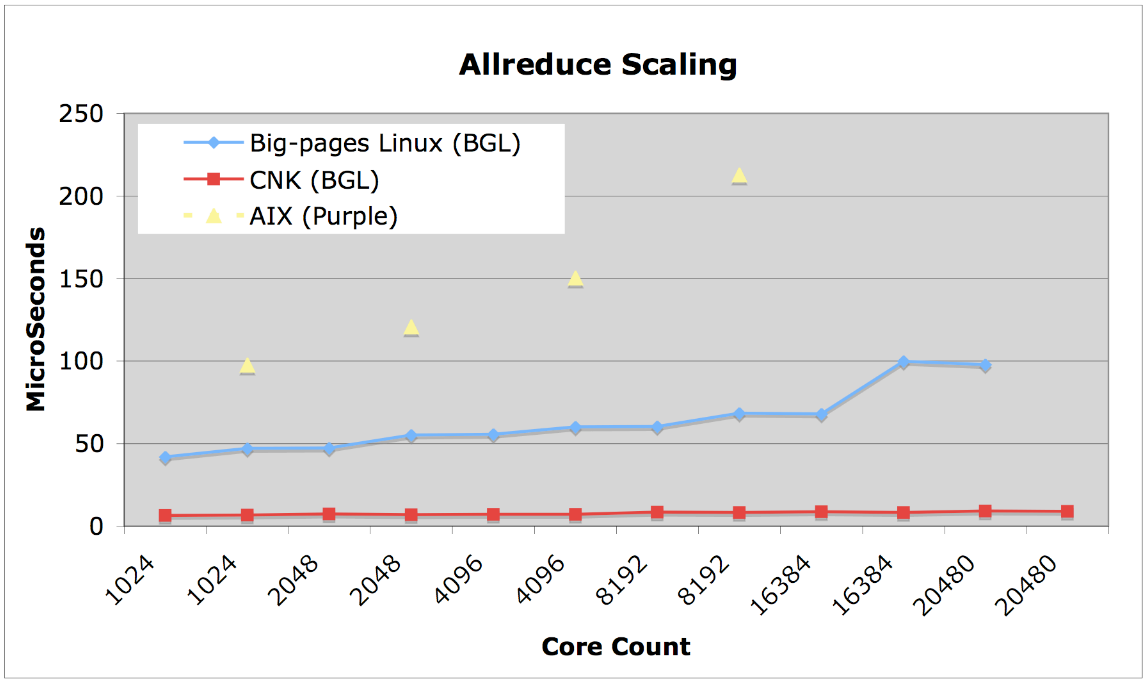

Figure 2: Allreduce scaling of a full Linux kernel (Big-pages Linux) on a 20-rack BlueGene system

4. Conclusion & Final Comments

Our experimental results on BGW enabled us to verify that our approach of full Linux for large systems such as Blue Gene is pragmatic enough to be a valid option. Unless we aborted a test (sometimes done for long-running jobs), the tests completed execution in a normal fashion. The results that we obtained indicate that boot times for diskless full Linux gradually increase to around 14 minutes at 20,480 nodes for BGW type architectures. Furthermore, Allreduce scaling of our Big-pages Linux kernel scales much better than some other existing platforms, but CNK measurements indicate there is room for improvement. We plan future testing of our parallel-aware and lock optimization techniques to redress this point. Overall, the opportunity to conduct these tests on a large machine such as BGW was a very valuable step in our research.

Acknowledgements

We are very grateful to IBM for offering us time on BGW and for giving us the opportunity to test our codes on the full system. In particular, we thank Fred Mintzer and his team at IBM for the support provided during BGW-Day. We would also like to thank the author of the Presta Suite – Chris Chambreau.

5. References

[Chakravorty06] Sayantan Chakravorty, Celso L. Mendes, Laxmikant V. Kalé, Terry Jones, Andrew Tauferner, Todd Inglett and José Moreira. HPC-Colony: Services and Interfaces for Very Large Systems. ACM SIGOPS Operating Systems Review: Operating and Runtime Systems for High-end Systems, 40(2), April 2006.

[Gupta91] A. Gupta, A. Tucker, and S. Urushibara, "*The Impact of Operating System Scheduling Policies and Synchronization Methods on the Performance of Parallel Applications*," In Proceedings of the 1991 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pages 120-132, May 1991.

[Jones03a] Terry Jones, Jeff Fier, Larry Brenner, Observed Impacts of Operating Systems on the Scalability of Applications. LLNL Technical Report UCRL-MI-20269, March 5, 2003.

[Jones03b] Terry Jones, Shawn Dawson, Rob Neely, William Tuel, Larry Brenner, Jeff Fier, Robert Blackmore, Pat Caffrey, Brian Maskell, Paul Tomlinson, Mark Roberts, Improving the Scalability of Parallel Jobs by adding Parallel Awareness to the Operating System. Proceedings of Supercomputing 2003, Phoenix, AZ, November 2003.

[Jones07] Terry Jones, Andrew Tauferner, Todd Inglett. HPC System Call Usage Trends, the 8th LCI International Conference on High Performance Computing, South Lake Tahoe, CA, May 2007.